

SN3262 – Network Administration, Management and Security

Exercise 0: Basic Unix/Linux commands

Objective

- To investigate some of the Unix/Linux commands
- To use the man pages

UNIX and Linux commands are case-sensitive. **LS** is not a recognised command; **ls** is. File names are also case-sensitive.

/ is used to separate path names (“Windows” uses \)

The form of UNIX or Linux commands is < command > < modifiers > < object >.

e.g. ls -l *.txt

Gives long listing of all files in the current directory with the 'txt' file extension

pwd

```
print working directory
```

This will tell you what directory you are in. This can be very useful as you may have several terminal windows open at once. It is also useful to keep track of where you are if you have used relative pathnames. This version of Linux shows the working directory in the prompt.

cd

change directory

cd followed by no path changes to your own top directory

Navigation of the directory tree

The directory tree can be navigated using relative path names

.. up a level

. here

examples:

```
cd .. up one level
```

```
cd ../.. up two levels
```

1. Login to Linux. Open a terminal window. Write down your directory name. Now navigate up the tree using `cd`, sketching the directory tree and writing down the names of the nodes. When you have reached the root of the tree, return to your own top directory.

On-line reference for Linux commands

man < **command** > will show the manual pages for a given command.

man -k < keyword > will show any header line of a man page that contains one of the specified keywords.

2. Type **man ls** to see the man pages for the **ls** command. Quickly scroll through them to get an idea of the layout and format used.

ls

examples

- ls** **ls** without an object will list all the files in the current directory
- ls -l** listing long. This lists details of all the files in the current directory
- ls -a** listing all
- ls -al** long listing of all files
- ls -la** long listing of all files
- ls -t** listing ordered by time of modification (newest first)
- ls -r** list files in reverse order (by name or by time)

3. Note the difference between **ls** and **ls -a** in your directory.

mkdir

make subdirectory

mkdir < subdirectory >

4. Make a subdirectory in your directory called **ex0**. Change to **ex0** and make a subdirectory **catfiles**. Change to this directory. Use **pwd** to find out where you are.

cat

create files

concatenate files

list on the screen

5. Look at the man pages for **cat**.

Redirection; stdin and stdout; pipes

In Linux and Unix, the standard input, **stdin**, is the keyboard and the standard output, **stdout**, is the VDU screen.

Redirection is indicated by the operators **<** and **>** where **<** is “where from” and **>** is “where to”. The output of one command can be “piped” to another. **|** is used for this.

6. Use **cat** to create a file, **filex**, by typing **cat > filex**. Type in some lines. Use **Ctrl-D** to terminate the input to the file. Do the same to create a file **filey** and a file **filew**. Try the following commands and make a brief note of what they do:
cat filex

cat filey filex

cat filex filey

cat filex >> filey This will append **filex** to **filey**. Look at **filey**.

What would happen with

cat filex > filey

cat filey filex > filez

ls|cat > filep|ls ?

Wildcards

* matches a number of characters

? matches a character

7. Try **ls *z**.

more **more < filename >**

“q” to quit

less

less < filename >

“q” to quit

chmod

change mode

This will change the permissions of a file. In **ls -l**, you will see at the left of the file name for example **rw-r--r--**. The first three characters represent the user permissions, the next three the group and the final three the “rest of the world”. This set would indicate that the owner had read, write and execute permission, and that the group and other had only read access.

u user (i.e. owner)

g group

o other (rest of the world)

a all (this is default)

The permissions can be given using octal representation with 4 for read, 2 for write and 1 for execute. You can also use “+” to add permissions, “-” to remove and “=” to set permissions.

8. Make a file **filet**. What permissions does it have?

What permissions would be set for **filet** using **chmod 751 filet**?

And using **chmod 444 filet**?

If **filet** has permissions 444

what will it have after **chmod u+x filet** ?

what will it have after **chmod u=rwx,g=rx,o=x filet** ?

what will it have after **chmod a-wx,a+r filet** ?

what will it have after **chmod =r file** ?

cp

copy a file

cp < file1 > < file2 >

Note: not copy!

9. Copy the file filez to filer. Copy the file filer to ex0 using relative path names. Use the **ls** command without changing directories to check that the copy has succeeded. Change to your top directory. Copy the file filer from catfiles using “here”.

10. **whereis**

whereis < name >

Use **whereis** to find out where the file **ifconfig** is.

11. **history**

history 3 show the last 3 commands

!! repeat last command

!3 repeat third command

12. **mv**

move a file

Change back to your directory ex0/catfiles. Look at filer and flew. Now execute

mv flew filer

What has happened to the information held in filer?

And to the information in flew?

13. **rm**

remove a file

rm filer

Use **ls** to check the removal. Note that there is no “are you sure” approach in Unix or Linux.

Contemplate WITHOUT DOING IT the effect of **rm ***.

14. **grep**

grep searches files for lines matching a given pattern.

grep < options > < pattern > < file >

options

-i ignore upper and lower case distinctions

-c print only a copy of matched lines

-n print lines and their line numbers

patterns

[] match one from a set

example

[Tt] match containing T or t

[R-T,r-t] match containing the range R-T or the range r-t

Use **grep** to search one of the files you have created for an expression. Then use it to search several files by name. Then use it to search several files using wildcards for the names.

You should download a copy of “An Introduction to the Linux Command Shell For Beginners” from <http://vic.gedris.org/Manual-ShellIntro/1.2/ShellIntro.pdf> and “Bash Guide for Beginners” from <http://www.tldp.org/guides.html>

Two other useful guides are

<http://linuxcommand.org/lts0010.php>

<http://steve-parker.org/sh/sh.shtml>

You will find all of the above useful as references throughout the course.